

Japanese Patent Laid-open Publication No.: SHO58-106636

Publication date : June 25, 1983

Applicant : HITACHI, LTD.

Title : PIPELINE OPERATING DEVICE

5

Fig. 3 is one embodiment of the present invention. Reference numerals 1 to 20, and 22 denote like parts to those shown in Fig. 1, and show a 3-stage pipeline calculation device which is provided with three calculating circuits 2, 3, and 4 corresponding to each stage. Behind an instruction register 10 20 that receives control information (instruction) from the instruction reading circuit 18, instruction registers 23, 24, and 25 corresponding to each of the stages and the calculating circuits are provided in series, and reference numerals 30, 31, and 32 denote set signals for each of the instruction registers 23, 24, and 25, respectively. Instruction decoders 26, 27, 28, and 29 are 15 connected to the instruction registers 20, 23, 24, and 25, respectively. The instruction decoders 26, 27, 28, and 29 decode control information within their respective instruction registers 20, 23, 24, and 25, and instruct the corresponding calculating circuits to make calculations in accordance with the control information. For example, when the control information indicates an 20 instruction of floating-point calculation, the calculating circuit 2 makes a pre-normalizing operation, the calculating circuit 3 makes a calculation of a significand (addition or subtraction), and the circulating circuit 4 makes a post-normalizing operation. When the control information indicates an instruction of fixed-point calculation, the calculating circuits 2 and 4 are not 25 operated, but calculation is made by the circulating circuit 3. In the case of

other calculations, the calculating circuit 2 makes a shifting operation.

⑨ 日本国特許庁 (JP)

⑩ 特許出願公開

⑫ 公開特許公報 (A)

昭58—106636

⑮ Int. Cl.³
G 06 F 7/00
9/38

識別記号

庁内整理番号
7313—5B
6745—5B

⑬ 公開 昭和58年(1983)6月25日

発明の数 1
審査請求 有

(全 6 頁)

⑭ パイプライン演算装置

⑯ 特 願 昭56—203702

⑰ 出 願 昭56(1981)12月18日

⑱ 発 明 者 畠山靖彦

秦野市堀山下1番地株式会社日
立製作所神奈川工場内

⑲ 発 明 者 村山浩

秦野市堀山下1番地株式会社日
立製作所神奈川工場内

⑳ 出 願 人 株式会社日立製作所

東京都千代田区丸の内1丁目5
番1号

㉑ 代 理 人 弁理士 薄田利幸

明 細 書

1 発明の名称 パイプライン演算装置

2 特許請求の範囲

(1)、演算を複数のステージに分割し、ステージ毎にオーバーラップして処理するパイプライン演算装置において、各々が上記ステージに対応して設けられた複数の演算回路と、各々が上記個々の演算回路又は複数の演算回路に対応して設けられ、演算内容を指示する制御情報を保持する複数のレジスタと、該レジスタが保持する制御情報を直接あるいはデコードして対応の演算回路に与えて演算を指示するための手段とからなり、上記各レジスタ内の制御情報および各演算回路の出力をそれぞれ順次次のステージのレジスタおよび演算回路に与えることを特徴とするパイプライン演算装置。

(2)、上記レジスタは演算回路対応に設けられることを特徴とする特許請求の範囲第1項記載のパイプライン演算装置。

(3)、上記複数のレジスタの前段にさらに制御情報を保持するレジスタを有し、該前段のレジスタ内の制御情報はオペランドベクトルの要素数に等しいサイクルの間保持され、後段の各レジスタは各サイクル毎に前のステージのレジスタの内容に更新されることを特徴とする特許請求の範囲第1項記載のパイプライン演算装置。

(4)、上記オペランドベクトルの要素対応に有効性を示すパリティビットを付すようにし、このパリティビットの保持手段を上記レジスタ対応に設けて同じように順次次のステージの保持手段に移すようにし、このパリティビットによって対応のレジスタへの制御情報の設定を制御することを特徴とする特許請求の範囲第3項記載のパイプライン演算装置。

3 発明の詳細な説明

発明の対象

本発明はパイプライン演算装置に関する。さ

らに詳細には複数の種類の演算をオーバーラップして処理することのできるパイプライン演算装置に関する。

従来技術

パイプライン演算装置は、演算を複数のステージに分割し、ステージ毎にオーバーラップして処理する。各ステージに対応して演算回路が設けられ、各演算回路は演算指示に応じた演算を行ない、出力は次のステージの演算回路に与える。従来のパイプライン演算装置は一種類の演算を複数組の入力データに対して連続的にオーバーラップしつつ処理することができるが、異なる演算を連続して処理することはできず、一つの種類の演算が終了するまでは、次の異なる種類の演算をオーバーラップして開始することができない。例えば、浮動小数点の加算を複数組の入力データに対して連続的にオーバーラップしつつ実行することはできる。しかしながら、浮動小数点の加算と固定小数点の加算あるいは浮動小数点の減算などの異なる種類の演算を連続的

にオーバーラップしつつ実行することはできなかった。

第1図は従来の典型的なパイプライン演算装置を示す。ここでは3ステージからなるパイプライン演算装置を示し、3つの演算回路2,3および4を含む。この装置のデータ系は、入力オペランドレジスタ1、中間ステージラッチ5および6、演算結果用レジスタ7、および演算回路2,3および4からなる。記憶装置8から送出されたオペランドはデータ入力バス9を介して上述のデータ系に入り、演算結果は結果の書き込みバス10を介して記憶装置8に格納される。記憶装置8としては、計算機システムの主記憶装置でも、またデータレジスタ群であってもよい。

一方、演算内容を指示するオペレーションコードあるいはオペレーションコードに相当する情報（制御情報という）は命令脱出回路18から命令起動バス19を介して命令レジスタ20にセット信号22によってセットされる。命令レジスタ20にセットされた制御情報はデコード21でデコ

ードされる。デコード21は制御情報に応じて、データ系に対する制御信号、即ち、ラッチのセット信号11~14および演算制御信号15~17を発生する。

第2図は第1図のパイプライン演算装置で2種類のベクトル演算AおよびBを連続して行なう場合のタイムチャートを示す。演算A,Bは異なる演算内容を指示している。第2図では、演算A,B共に3組の入力データに対して行なう場合を例として示しており、①~④で示すのがオペランドベクトルの要素番号を表わす。第2図から明らかなように、演算Aが行なわれる要素①, ③, ⑤は連続的にオーバーラップしつつ実行されるが、演算Aの最後の要素⑥が入力されてもその結果が出力されるまで命令レジスタ20の内容を変更できず、そのため次の演算Bの最初の要素①を入力するまでに3サイクルの無駄サイクルが生ずる。

発明の目的

本発明の目的は複数の異なる種類の演算をオ

ーラップして処理することができるパイプライン演算装置を提供することにある。

発明の総括的説明

本発明は各ステージに対応して演算回路があり、この個々の演算回路又は複数の演算回路に対応して演算内容を指示する制御情報を保持するレジスタが設けられる。このレジスタが保持する制御情報は直接あるいはデコードして対応の演算回路に与えられ、演算を指示する。そして各レジスタ内の制御情報および各演算回路の出力はそれぞれ順次次のステージのレジスタおよび演算回路に与えられる。

発明の実施例とその効果

第3図は本発明の一実施例を示す。1~20および22は第1図の同符号と同部分を示しており、同様に3ステージからなるパイプライン演算装置を示し、各ステージに対応した3つの演算回路2,3および4が設けられる。命令脱出回路18からの制御情報（命令）を受ける命令レジスタ20の後には各ステージ、演算回路対応の命令レ

ジスタ 23, 24 および 25 が直列に設けられ、それぞれのセット信号が 30, 31 および 32 で示される。命令レジスタ 20, 23, 24 および 25 の各々には命令デコーダ 26, 27, 28 および 29 が接続される。命令デコーダ 26, 27, 28 および 29 はそれぞれ命令レジスタ 20, 23, 24 および 25 内の制御情報をデコードし、対応の演算回路に制御情報に応じた演算を指示する。例えば制御情報が浮動小数点演算を指示していれば、演算回路 2 はブレンーマライズ動作、演算回路 3 は仮数部の演算（加算あるいは減算等）、演算回路 4 はポストノーマライズ動作を行ない、固定小数点演算を指示していれば、演算回路 2 および 4 は動作されず、演算回路 3 で演算を行なう。また他の演算では演算回路 2 はシフト動作を行なう。

またレジ 1, 7、ラッチ 5, 6 の開閉、セット等を指示する。例えばデータ入力バス 9 は第 1 オペランド、第 2 オペランドのためのバスを有しており、両オペランドのセット、一方のみのオペランドのセット等の指示も行なう。命令レジ

スタ 23, 24 および 25 が直列に設けられ、それぞれのセット信号が 30, 31 および 32 で示される。命令レジスタ 20, 23, 24 および 25 の各々には命令デコーダ 26, 27, 28 および 29 が接続される。命令デコーダ 26, 27, 28 および 29 はそれぞれ命令レジスタ 20, 23, 24 および 25 内の制御情報をデコードし、対応の演算回路に制御情報に応じた演算を指示する。例えば制御情報が浮動小数点演算を指示していれば、演算回路 2 はブレンーマライズ動作、演算回路 3 は仮数部の演算（加算あるいは減算等）、演算回路 4 はポストノーマライズ動作を行ない、固定小数点演算を指示していれば、演算回路 2 および 4 は動作されず、演算回路 3 で演算を行なう。また他の演算では演算回路 2 はシフト動作を行なう。

またレジ 1, 7、ラッチ 5, 6 の開閉、セット等を指示する。例えばデータ入力バス 9 は第 1 オペランド、第 2 オペランドのためのバスを有しており、両オペランドのセット、一方のみのオペランドのセット等の指示も行なう。命令レジ

スタ 23, 24 および 25 のセット信号は各サイクル毎に発生回路 41, 42 および 43 から発生される。従って命令レジスタ 23~25 内の制御情報は毎サイクル更新され、その内容は次ステージに対応する命令レジスタに移される。従って、各ステージ毎に異なる制御情報を保持して、各々演算回路を制御することができるので、演算 A の最後の組の入力データが入力オペランドレジスタ 1 にセットされるのと同時に、演算 B の制御情報を命令レジスタ 20 にセットし、すぐ次のサイクルで演算 B の最初の組の入力データを入力オペランドレジスタ 1 にセットすることにより、パイプライン演算装置の全てのステージを有効に利用することができる。第 4 図を参照すると、3 サイクルに亘って A, B 2 種の演算が同時に、オーバーラップして実行されていることがよく理解できる。

第 3 図において、命令レジスタ 23~25 は各ステージに対応して設けられているが、実行する

スタの制御情報をデコードすることなくそのまま演算回路やレジスタ、ラッチに与えて制御可能であれば、デコーダ 26~29 は必要ない。40~43 は命令レジスタへのセット信号 22、30~32 を発生する発生回路である。どのようなセット信号を発生するかは第 4 図で説明される。

第 4 図は第 3 図のパイプライン演算装置で 2 種類のベクトル演算 A および B を連続して行なう場合のタイムチャートを示す。第 4 図では第 2 図と同様、演算 A, B 共に 3 組の入力データに対して行なう場合を例として示し、①~④がオペランドベクトルの要素番号を表わす。

命令レジスタ 20 のセット信号 22 は要素数に等しいサイクルで発生回路 40 から発生される。この場合要素数は 3 であり、3 サイクル間を置いて発生されている。この要素数はベクトル長レジスタ（図示せず）で示され、発生回路 40 はこのベクトル長レジスタの内容に基づいて、命令レジスタ 20 にセット信号を発生する。従って命令レジスタ 20 はその 3 サイクルに亘って更新され

演算の種類の変更において必ず 1 サイクルの空きを許すならば、これらの命令レジスタを、連続した 2 ステージに対応させ、命令レジスタはその 2 つの演算回路に共通に設けることも出来る。

第 3 図、第 4 図における実施例では、演算に使用するオペランドベクトルが記憶装置 8 から自由に読出せる場合、即ち命令起動に同期してオペランドの読出しが開始し、各要素は連続して読出せる場合について示した。

これに対して、オペランドベクトルが要素単位に飛び飛びに読出され、パイプライン演算器に対して各要素が必ずしも連続して入力されない場合もある。例えばオペランドベクトルがパイプライン演算器に供給される段階に至ったが、要素によっては未だ記憶装置 8 に図示されない装置から供給されてきていない場合である。このような場合には記憶装置 8 から、データ入力バス 9 上のデータの有効性を示すパリティビットを出力することにより制御することがで

きる。以下にバリデイティビットが対応する要素より1サイクル前に出力される場合を例として、オペランドベクトルの各要素が必ずしも連続的に供給されない場合の実施例を説明する。

第5図を参照するに、記憶装置8から出力されたバリデイティビット33は、発生回路44から発生されるセット信号38によって毎サイクルたかかれるバリデイティビット用フリップフロップ34~37により順次取り込まれ、シフトされるこのフリップフロップ35~37の出力30~32によってステージ対応の命令レジスタ23~25をセットする。バリデイティビットが'0'のとき、フリップフロップはセットされないで、命令レジスタ23~25へのセット信号30~32も出力されず、命令レジスタ23~25には制御情報が設定されない。またフリップフロップ34~37の出力はANDゲート39に与えられており、'0'のときはレジスタ、ラッチへのセット信号を阻止する。その他の構成においては第3図と同様である。

第6図は第5図のタイムチャートである。第

6図において、バリデイティビット33は演算Aの第4番目、演算Bの第3番目において'0'であり、その要素に対しての演算は行なわれていないが、それ以外においては第3図、第4図と同様な動作である。

以上はベクトル演算について説明したが、第3図に示したパイプライン演算装置を汎用計算機においても用いることが可能である。第7図は第3図のパイプライン演算装置を汎用計算機に用いた場合のタイムチャートである。第7図から明らかなごとく、セット信号は発生回路40~43から毎サイクル発生され、全ての命令レジスタ20,23,24,25は毎サイクル更新される。第7図ではA~Eで示す5種類の演算が行なわれる様子を示し、これらが連続してオーバーラップして処理されていることを示している。

発明の効果

本発明によれば、演算回路がステージに対応して設けられると共に、制御情報を保持するレジスタが個々の演算回路又は複数の演算回路に

対応して設けられ、レジスタが保持する制御情報によって対応の演算回路に演算を指示するようになったので、複数の種類の演算をオーバーラップして処理することができる。

4 図面の簡単な説明

第1図は従来例を示すブロック図、第2図は第1図を説明するタイムチャート、第3図は本発明の一実施例を示すブロック図、第4図は第3図がベクトル演算のために動作する場合を説明するブロック図、第5図は本発明の他の実施例を示すブロック図、第6図は第5図を説明するタイムチャート、第7図は第3図を汎用計算機に用いた場合のタイムチャートである。

1…入力オペランドレジスタ、

2~4…演算回路、

5および6…中間ステージデータラッチ、

7…演算結果用レジスタ、

8…記憶装置、

20,23,24,25…命令レジスタ、

26~29…命令デコード、

34~37…バリデイティビット用フリップフロップ、

40~44…セット信号発生回路。

図 1

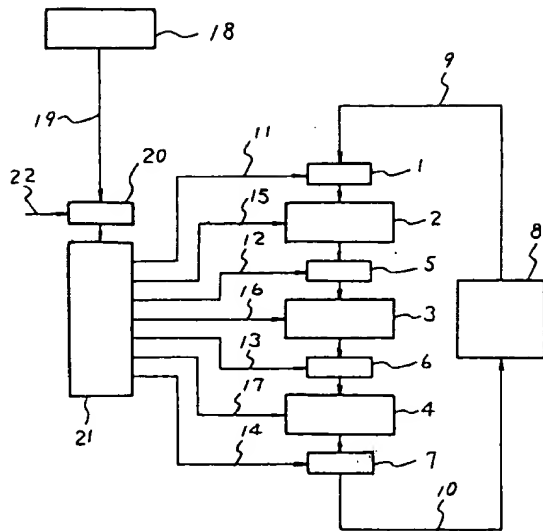


図 2

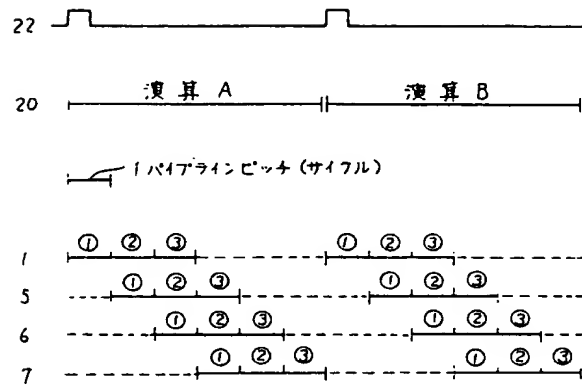


図 3

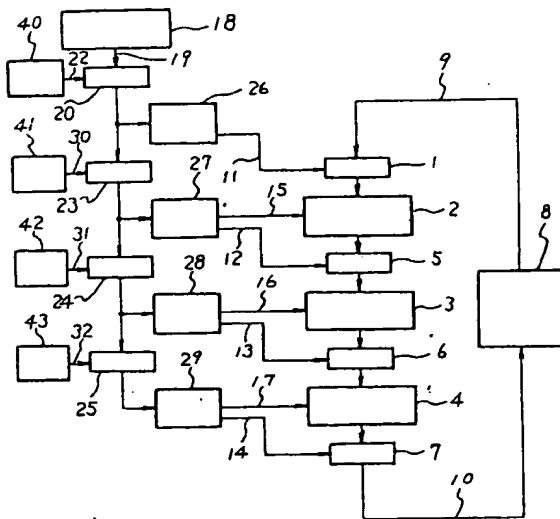


図 4

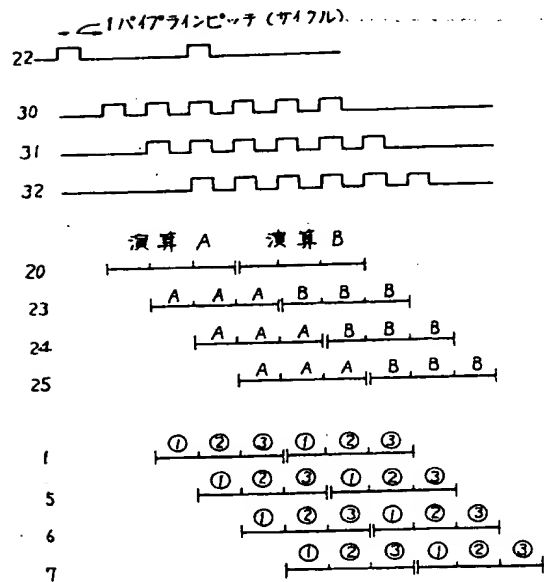


図 5

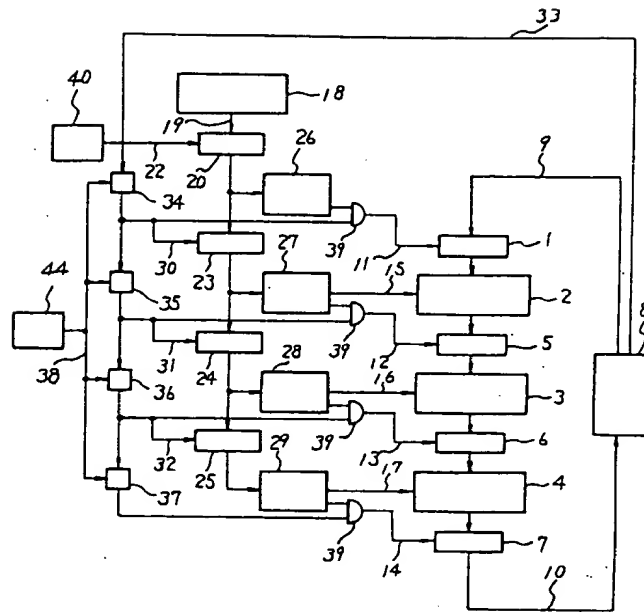


図 6

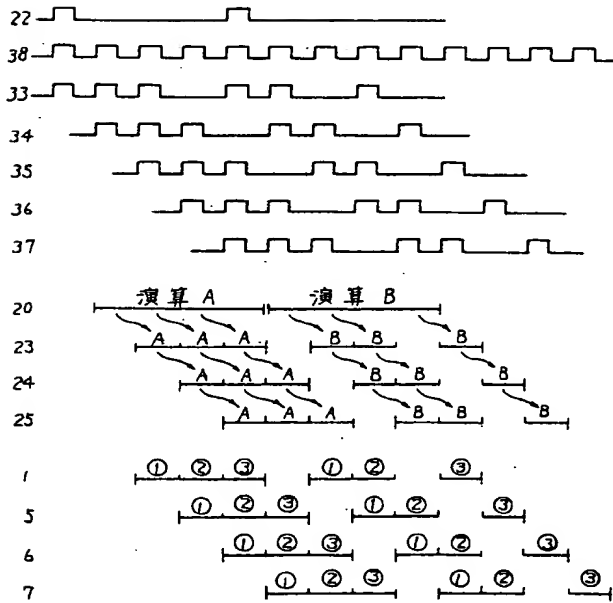


図 7

